In the Claims

Please amend claims 8, 15, 23, and 31 as follows.

1. (Original) A method for incrementally scaling a file system, comprising:

adding a new file storage device to a file system having a storage space comprising at least one other file storage device having a plurality of directories and files stored thereon to form a new file system configuration; and

migrating a portion of the files from said at least one other file storage device to the new file storage device while hiding such migration from client applications that access files from the file system so as to not affect file access operations requested and performed by the client applications during the migration.

2. (Original) The method of claim 1, wherein the portion of files that are migrated from said at least one storage device to the new storage device is selected such that the files are distributed across all of the storage devices in the file system after the migration is completed based on a relative capacity of each of the storage devices in the system.

3. (Original) The method of claim 1, wherein the file storage devices are accessed using a file system protocol, further comprising providing a storage abstraction layer between the client applications and the file system protocol, said storage abstraction layer providing an interface to the client applications that presents the file system as a virtual file system.

4. (Original) The method of claim 3, further comprising providing information corresponding to the new file system configuration to the storage abstraction layer.

5. (Original) The method of claim 3, wherein the storage abstraction layer distributes new files created by the client applications across all of the storage devices in the file system so as to load balance access operations of the files.

6. (Original) The method of claim 3, further comprising:

filtering requests made by client applications to access a file stored on the file system, said requests referencing a virtual storage location of the file; and

remapping the file access requests that are filtered from the virtual storage location to a physical location on a storage device on which the file is actually stored; and

accessing the file through use of the file system protocol by referencing the physical location of the file.

7. (Original) The method of claim 1, wherein migrating the files to the new storage device comprises:

identifying a source location corresponding to a storage device and directory in which each file is initially stored;

identifying a destination location for each file corresponding to a directory on the new storage device the file is to be stored in;

copying each file from the source location to the destination location;

deleting each file from its source location;

monitoring for any file access requests made by an client application while the file is being migrated; and

aborting the migration of the file if a file access request is made during the migration of the file.

8. (Currently Amended) The method of claim 7, further comprising putting any file whose migration is aborted into a queue such that the migration of such file may [[by]] be retried at a future time.

9. (Original) The method of claim 7, further comprising:

providing a lock on each file during its migration; and

allowing the lock to be stolen by a client application if the client application requests access to the file during its migration.

10. (Original) The method of claim 7, further comprising:

providing a lock token for each file opened by a client application, said token identifying that its corresponding file is currently in use and not available to be migrated.

11. (Original) The method of claim 7, when each token is assigned an expiration time after which the token is no longer valid.

12. (Original) The method of claim 11, further comprising:

putting a file having an unexpired token into a queue such that the migration for such file may be retried at a future time; and

migrating the file after the token has expired.

13. (Original) The method of claim 1, further comprising:

partitioning the storage space of the file system into fragments; and

assigning files in the file system to corresponding fragments.

14. (Original) The method of claim 13, wherein the files are assigned to corresponding fragments based on the directories the files are in.

15. (Currently Amended) The method of claim 13, wherein the directories are assigned to corresponding fragments in a ~~substantially~~ random manner.

16. (Original) The method of claim 13, further comprising selecting (a) set(s) of fragments to be migrated when a new storage device is added to the system.

17. (Original) The method of claim 16, wherein the set(s) of fragments that are selected comprise a portion of a total number of directories on all of the storage devices in the file system such that after the set of fragments are migrated, each storage device has a proportionate amount of directories based upon its relative capacity.

18. (Original) The method of claim 1, further comprising:

providing an administrative tool that enables a user to add a new storage device to the configuration of the file system; and

automatically selecting the portion of files to be migrated to the new storage device based on the new configuration.

19. (Original) The method of claim 1, wherein the file system comprises a virtual volume corresponding to storage space provided by at least one storage device, said virtual volume including a plurality of virtual directories in which virtual files may be stored and having configuration data stored on the file system that maps virtual directories to physical directories.

20. (Original) The method of claim 19, wherein the configuration information comprises a master directory stored on a storage device, said master directory including a plurality of subdirectories, each corresponding to a respective virtual directory and

having an encoded pointer that points to a location on the file system where files corresponding to the virtual directory are physically stored.

21. (Original) The method of claim 20, wherein the configuration information further comprises a fragment map that identifies what storage device a directory and its files are stored on based upon the fragment(s) the directory is assigned to.

22. (Original) A method for load balancing file access on a network file system having a storage space provided by a plurality of network storage devices in which a plurality of files are stored, comprising:

partitioning the storage space into a plurality of fragments, each fragment being mapped to one of said plurality of network storage devices;

assigning files among said plurality of files to fragments such that each fragment, on average, comprises a substantially equal number of files;

migrating files among said plurality of files from network storage devices on which they are initially stored to other network storage devices corresponding to the fragment they are assigned to in a manner such that the migration of files are undetectable to client applications that access the network file system.

23. (Currently Amended) The method of claim 22, further comprising assigning new files that are created by the client applications to fragments on a ~~substantially~~ random basis.

24. (Original) The method of claim 22, wherein each file is assigned to its corresponding fragment based upon the directory the file resides in.

25. (Original) The method of claim 24, further comprising splitting directories into a plurality of portions, wherein each directory portion of files is assigned to a respective fragment.

26. (Original) The method of claim 22, further comprising providing a storage abstraction layer that enables the client applications to access the network file system as a virtual storage space including at least one virtual volume comprising a plurality of virtual directories and file names.

27. (Original) The method of claim 26, further comprising providing the storage abstraction layer with access to a fragment map that maps each fragment to a storage device to which the fragment is hosted.

28. (Original) The method of 27, wherein each virtual directory has a corresponding physical directory on one of said plurality of network storage devices, and wherein each virtual volume includes data stored on a network storage device that links each virtual directory to its corresponding physical directory.

29. (Original) The method of claim 28, wherein the data that links the virtual and physical directories comprises a master directory that includes a plurality of subdirectories stored on a network storage device, each subdirectory being named based on a corresponding virtual directory name and including at least one file having a name comprising indicia that identifies the location of the physical directory on the network file system corresponding to the virtual directory.

30. (Original) The method of claim 29, wherein said indicia pointer comprises a first portion that identifies the fragment the files are assigned to and a second portion identifying a name of the physical directory in which the files are stored.

31. (Currently Amended) A network file system comprising:

a plurality of file storage devices connected to a network;

at least one client machine having a processor, connected to the network in communication with said plurality of file storage devices, operating under an operating system running on the processor and ~~running~~ hosting a client application that runs on the operating system and accesses files stored on the network file storage system;

a storage abstraction layer comprising at least one module running on the processor and providing an interface to the client application that virtualizes the network file system such that it appears as a virtual storage space comprising a set of virtual directories and files to the client application; and

mapping data, accessible to the storage abstraction layer, that maps the virtual directories and files to corresponding physical directories and files stored on said plurality of file storage devices.

32. (Original) The network file system of claim 31, wherein the operating system provides the client application with access to files stored on the network file system via a set of calls implemented using a CIFS (Common Internet File System) protocol, and wherein the storage abstraction layer comprises a filter driver that intercepts calls to the network file system that reference virtual directories and virtual file names and remaps the directories and file names in such calls so that they reference the physical directories and file names to which the virtual directories and virtual file names correspond, based on the mapping data.

33. (Original) The network file system of claim 32, wherein the storage abstraction layer further includes an agent module running on the operating system and in communication with the filter driver that maintains a copy of the mapping information and forwards appropriate mapping information to the filter driver.

34. (Original) The network file system of claim 31, wherein the operating system provides the client application with access to files stored on the network file system via a set of calls implemented using an NFS (Network File System) protocol, and wherein the storage abstraction layer comprises at least one NFS daemon that intercepts calls to the network file system that reference virtual directories and virtual file names and remaps the directories and file names in such calls so that they reference the physical directories and file names to which the virtual directories and file names correspond, based on the mapping data.

35. (Original) The network system of claim 34, wherein the storage abstraction layer further includes an agent module running on the operating system and in communication with said at least one NFS daemon that maintains a copy of the mapping information and forwards appropriate mapping information to said at least one NFS daemon.

36. (Original) The system of claim 31, wherein the virtual file system comprises at least one virtual volume that is mapped to a set of physical directories and files on at least one of the file storage devices, further comprising:

a master directory stored on one of the file storage devices that includes a plurality of subdirectories, each being named based on a corresponding virtual directory name and including at least one file having a name comprising indicia that identifies the

location of the physical directory on the network file system corresponding to the virtual directory name.

37. (Original) The system of claim 36, wherein the indicia comprises a first portion that identifies the fragment the files are assigned to and a second portion identifying a name of the physical directory in which the files are stored.

38. (Original) The system of claim 31, further comprising:

an administrative tool running on one of said at least one client machines or another machine connected to the network, which enables a user to define an original or new configuration of the file system and provides configuration information concerning the original or new configuration of the file system defined by the user to other software components in the system.

39. (Original) The network file system of claim 31, further comprising:

a migration management module that facilitates a migration of files from a source file storage device on which the files are initially stored to a destination file storage device corresponding to where the files are stored after the migration.

40. (Original) The network file system of claim 39, further comprising:

a lock manager module that manages locks on files being migrated to ensure that the files do not become corrupted during a migration and automatically releases a lock on a file if it is requested to be accessed by a client application during a migration operation.